Multi-View Object Class Detection with a 3D Geometric Model

Jörg Liebelt, Cordelia Schmid

Multi-View Object Class Detection with a 3D Geometric Model

Joerg Liebelt
IW-SI, EADS Innovation Works
D-81663 Munich, Germany
joerg.liebelt@eads.net

Cordelia Schmid
LEAR, INRIA Grenoble
F-38330 Montbonnot, France
cordelia.schmid@inrialpes.fr

Abstract

This paper presents a new approach for multi-view object class detection. Appearance and geometry are treated as separate learning tasks with different training data. Our approach uses a part model which discriminatively learns the object appearance with spatial pyramids from a database of real images, and encodes the 3D geometry of the object class with a generative representation built from a database of synthetic models. The geometric information is linked to the 2D training data and allows to perform an approximate 3D pose estimation for generic object classes. The pose estimation provides an efficient method to evaluate the likelihood of groups of 2D part detections with respect to a full 3D geometry model in order to disambiguate and prune 2D detections and to handle occlusions. In contrast to other methods, neither tedious manual part annotation of training images nor explicit appearance matching between synthetic and real training data is required, which results in high geometric fidelity and in increased flexibility. On the 3D Object Category datasets CAR and BICYCLE [15], the current state-of-the-art benchmark for 3D object detection, our approach outperforms previously published results for viewpoint estimation.

1. Introduction

In recent years, multi-view generic object class detection has received increasing attention [3, 5, 12, 15, 16, 19]. Most approaches address the task by extrapolating known strategies from 2D single-view object class detection, notably by combining classifiers for separate viewpoints. Some authors have proposed to include weak geometric information into the learning process, mostly by applying locally deformable 2D models for discrete viewpoints [3, 5, 7]. Learning a generic representation of the 3D geometry of an object class, on the other hand, is challenging. While numerous 2D detection approaches have been developed which are capable of handling noise and large variation in intra-class appearance, the task of learning a robust 3D geometric model for an object class remains an active research topic [1, 15, 16].

The advantages of a 3D representation for multi-view object class detection are obvious: 2D part detections can be disambiguated and pruned with respect to their consistency with the object class geometry under full perspective projection, and detection confidence can be computed per-object instead of combining per-classifier scores. Furthermore, such a representation allows an approximate estimation of the pose. However, these advantages often come with an increased training complexity such as manual per-label annotations. More importantly, they usually cannot be flexibly integrated into existing 2D detectors. In contrast, this paper shows that a joint model for geometry and appearance can be avoided by learning separate models for both and combining them at a later stage. As a result, one can use better adapted, leaner representations and separate training sources and exploit the ubiquitous availability of geometrically faithful synthetic 3D CAD models for object detection tasks, while circumventing the gap between synthetic textures and real object appearance [12].

The paper is structured as follows. Section 2 summarizes previous work on multi-view object class detection. In section 3, an overview of the training approach is given. Details on the appearance model for hierarchical part-based detection on 2D training images are presented in section 4. The geometric representation of the object classes, which is built from synthetic 3D models, is described in section 5. Section 6 describes the combined detection process. Experimental results and a comparision with the state of the art are given in section 7 for the 3D Object Category datasets CAR and BICYCLE [15].

2. Related work

A survey of related work on multi-view object class detection shows three predominant approaches which differ in their choice of the geometric representation. 2D detectors can be combined by linking them over multiple viewpoints [17] and modeling flexible spatial layouts of part detectors [3, 5, 7]. Other methods have been proposed which build 3D representations of the object class from 2D train-

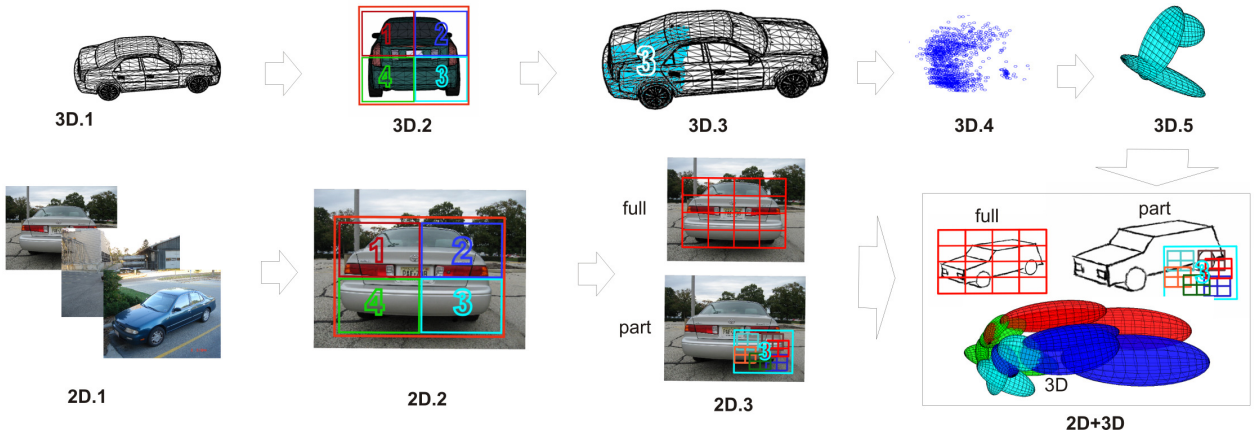

Figure 1. Overview of the two training steps. (Top) Mixture models are learnt from synthetic 3D models to describe the class geometry. (Bottom) Full object and part appearance are learnt from a 2D image database. See text for details. The figure is best viewed in color.

ing data based on initial viewpoint annotations [1, 9, 15, 16]. As a third approach, the use of existing 3D models has been suggested in the past [6] and more recently in [8, 12, 19].

The combination of 2D detectors to cover a multi-view sphere has been the initial step towards a more comprehensive use of geometry for object class detection [17]. In order to increase robustness towards pose changes, additional probabilistic layout models [3] as well as local 2D geometric constraints have been introduced in combination with increasingly powerful object part representations and learning procedures [5]. In [13], additional classifiers on spatial pyramids are learnt to obtain orientation estimates. However, these approaches are inherently limited to a few discrete viewpoints as detection output.

Alternatively, viewpoint-annotated training data can be used to dynamically build 3D representations to better address the possible viewpoint variations of object classes. In [15], homographic constraints between groups of object parts are combined to form a piecewise planar 3D approximation of object classes which also allows to interpolate unseen instances within the chosen parameterization. More recently, [16] introduced a probabilistic approach to learning affine constraints between object parts; during testing, they rely on a complex sequence of random forest classifiers, Hough voting and linear SVM classifiers. In [1], sparsely annotated 2D feature positions are factorized to obtain a 3D implicit shape model. Although these methods perform well, their training process is elaborate and they rely on relatively sparse object part representations which may impact their robustness for unseen objects.

The third category of multi-view object class detection approaches resorts to existing 3D models of varying level of detail. Initially suggested for example by [6] in the context of relational graphs, the idea has since been extended. Khan et al. [19] collect patches from viewpoint-annotated 2D training images and map them onto an existing 3D CAD model. Liebelt et al. [12] introduce a filtering step to build a 3D representation of both geometry and local feature appearance from a database of synthetic models. Heisele et al. [8] generate difficult training sets from synthetic 3D models for an active learning algorithm. While being significantly simpler to train, these methods suffer from rendering artifacts and a reduced similarity between synthetic models and real images.

In contrast, this paper presents a combination of the individual strengths of the above mentioned ideas, while circumventing their respective shortcomings. Based on discriminative part-based 2D detectors which are both robust and straightforward to train, only a few synthetic 3D models for each object class are used to learn a generative 3D representation of the object class geometry without relying on the presence of synthetic textures. In particular, no manual annotation of individual part locations is necessary. Moreover, a probabilistic pose estimation allows to obtain an approximate 3D object pose alongside a precise and robust 2D detection. This estimation step provides an effective evaluation measure to assess the consistency of the 2D part detections with respect to the full 3D geometry of the object class.

3. Overview

Figure 1 illustrates the approach presented in this paper, which is based on combining a 2D appearance model with an external 3D geometry.

The object class appearance is learnt from a database of 2D images, showing the objects from different viewpoints (figure 1, 2D.1); each image is annotated with the 2D bounding box and the viewpoint of the object, but neither manual part annotations nor segmentations are necessary. Each annotated bounding box is subdivided (2D.2) into a regular grid where each grid block represents a part of the object. A single spatial pyramid detector is used for the full object regions of interest (2D.3 top), while for each part region under each viewpoint, several smaller, overlapping

spatial pyramid detectors are trained (2D.3 bottom).

The 3D geometry is learnt from one or several synthetic 3D CAD models representative of the object class geometry. The models are rendered from many viewpoints (figure 1, 3D.1); the rendered images are subdivided (3D.2) into the same regular grid as in (2D.2). For each rendered pixel inside a part region, its original position on the CAD model surface is known; thus the image pixels belonging to the same part can be backprojected onto the surface (3D.3), sampled into discrete 3D points (3D.4) and the distribution of all 3D points belonging to one object part can be modeled by a mixture of Gaussians (3D.5).

The resulting object class representation now consists of a 2D pre-detector of regions of interest, dense 2D part detectors per viewpoint, and an approximate representation of the 3D geometry of the object class (figure 1, 2D+3D). To summarize, by subdividing both the annotated real training bounding boxes (figure 1, bottom) and the rendered images of the 3D models (figure 1, top) into the same regular grid of parts (figure 1, 2D.2 and 3D.2), the link between local 3D geometry and local 2D image appearance is established. Note that this requires bounding box annotations as well as approximative viewpoint annotations in the 2D training images.

4. Part-based Appearance Model

Learning the appearance of an object class needs to take into account large intra-class and viewpoint variations in addition to significant background clutter and partial occlusions. Moreover, when dealing with part-based object class detection, one aims at learning sufficiently powerful part descriptors for relatively small image patches. These patches do not always contain sufficient structure to be suitable for discriminative classifiers. In addition, manually including detailed annotations on the location of each object part is tedious. Consequently, some authors have suggested using fixed part layouts for 2D detection [2, 7, 14] where each detector is associated with an object part depending on its location inside the grid. More recently, the use of hierarchical structures as a representation for both the entire object and its subparts has been advocated [3, 5]. This work builds on these ideas in relying on spatial pyramids [11] both for the global object and the local parts. It extends beyond previous, sparse part-based approaches [14] by using both densely computed local features and spatial pyramids densely covering the image space. Learning the appearance of an object class consists of a two-fold supervised training process which is both efficient and robust; figure 1, 2D.3 illustrates the two detection components which are described in the following paragraphs.

4.1. Detector Layout

Both detection steps build on densely computed local features as their basic building blocks. The DAISY descriptor [18] was chosen because of its efficient implementation. Initially, from all positive and negative training images, DAISY descriptors are randomly sampled and clustered into a small codebook of fixed size C using a standard k-means algorithm with random initialization. The codebook size can be adapted to the complexity of the object class; see section 7 for details on the chosen parameters. Given each positive training annotation, DAISY features are then computed densely within the annotated training region and assigned to their respectively closest codewords to build localized occurrence histograms.

A single detector is trained on entire objects to identify regions which have a higher likelihood of containing an entire object instance; figure 1, 2D.3 top, shows an example layout. The dimensions and aspect ratio of the training annotations determine the dimension of the detector used during testing.

For the part-based detectors, instead of manually selecting semantic parts, the training annotations are further subdivided into a regular grid of $V \times W$ regions, assuming that the densely sampled detectors whose centers fall into the same region can be considered to share some common characteristics of this part of the entire object under this viewpoint; see figure 1, 2D.3 bottom. Consequently, $V \times W$ groups of detectors are obtained, each representing the appearance of a part of the entire object under the current view. These part detectors have to deal less with background variation, but focus primarily on differentiating between the appearance of areas of an object under changing viewpoints.

The negative training examples for object parts and full objects are initially chosen randomly on the background of the training images; the detector layouts which were used for the positive training instances are re-used to determine the layout of the negative samples.

4.2. Appearance Representation

Following [11], the localized occurrence histograms are combined into spatial pyramids. For the full object predetector, a single spatial pyramid is built to represent the appearance of the entire object under the current view; see figure 1, 2D.3 top. For the part-based detectors, $V \times W$ groups of spatial pyramids are obtained, each representing the appearance of a part of the entire object under the current view. The spatial pyramids of each part are densely sampled and allowed to overlap in order to completely cover the part area as shown in figure 1, 2D.3 bottom.

Given positive and negative training examples, separate SVM classifiers are now trained, one for the entire object under all viewpoints and one for each of the $V \times W$ object

parts under each of the weakly-annotated views as provided by the training database. In the case of the 3D Object Category datasets CAR and BICYCLE, annotations are given for discrete distances and elevation and azimuth angles (also see section 7). As illustrated in figure 2, the proposed approach parameterizes the viewpoints in spherical coordinates of $v = (r, a, b)$ where r is radius, a azimuth and b elevation, assuming a simplified camera which is always oriented at the centroid of the object. A part is assigned to one block of the fixed grid when its center falls into the block, thereby allowing for some overlap between the parts (see figure 1, 2D.3 bottom). All views having azimuth angles within a given range together with all distances and elevation angles associated with this azimuth angle are combined to train $V \times W$ part detectors for this particular base viewpoint; see section 7 for details on the chosen parameters. To compensate for the random choice of initial negative training instances, a standard bootstrapping procedure is used to iteratively select the most difficult false positives and false negatives for each part classifier. The SVMs are learned on a pyramid intersection distance kernel with the per-level weighting scheme suggested in [11].

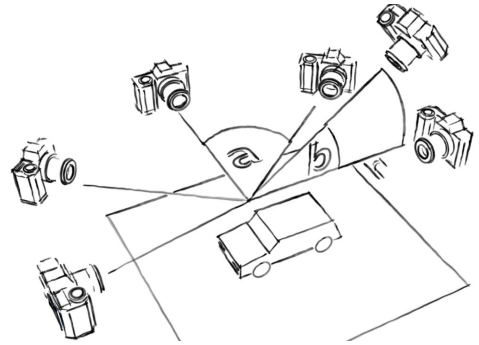

Figure 2. Discretization of the viewpoints for initial classification into "base viewpoints" in discrete azimuth steps, each combining multiple elevations and distances.

5. Geometry Model

The following section outlines how the model of the object class geometry is built to represent the 3D distribution of the centers for each of the $V \times W$ parts per object class and for each discretized camera viewpoint (figure 2).

Recently, some publications have proposed methods for building a 3D representation from trainingdata to be used for detection tasks. In most cases, groups of consistently deforming image regions are promoted to a higher geometry model to reflect their co-occurrence [1, 15, 16]. In [12], synthetic models with reduced textural similarity to real images have been used to compute filtered local features and to project their locations in rendered images into a common 3D coordinate system.

In this paper, a different approach is proposed which relies on commercially available synthetic 3D models; see figure 3 for some examples. However, unlike all previous approaches, the geometric learning task is separated from the appearance component. No explicit matching between synthetic textures and real images is required; still the precise geometry of synthetic models can be used in an extremely flexible way to learn the 3D distribution of parts of an object class, as long as the models represent characteristic object class geometries. In particular, no manual annotation of part locations is required. By limiting the contribution of the synthetic models to their geometry, far fewer models are needed to represent the geometry of an object class rather than all possible textural appearance variations.

5.1. 3D Training Data

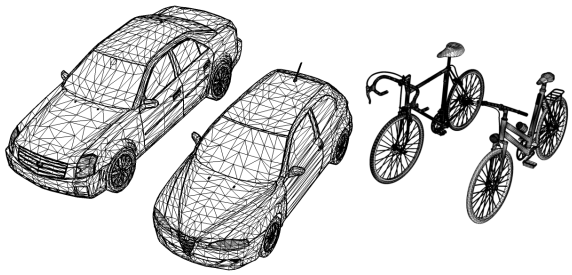

Figure 3. Synthetic 3D models used for the geometry training.

The use of synthetic models as training sources for the geometry allows to densely sample the space of possible viewpoints and to choose the models such that the training database includes representative object surface geometries. The approach follows the pose space parameterization of [15] as defined by their test database; the parameterization is based on a spherical coordinate system as illustrated in figure 2; a simplified camera model is assumed where rotations around the camera view axis are not part of the parameter space.

For each object class, all its 3D models are rendered into images of fixed dimensions, along with their automatically generated bounding boxes. Each model is rendered from the same viewpoints that are present in the real image database (termed "base viewpoints") as well as from additional densely sampled viewpoints, reflecting intermediate distances and object orientations. By using synthetic models, viewpoints can be more densely sampled from the space of all relevant poses to account for the typical visibility of the parts under perspective projection, depending on 3D surface structure and local self-occlusions. For each rendered view, the bounding box is subdivided into a regular grid of $V \times W$ parts (see figure 1, 3D.2) in the same way as for the appearance training (see figure 1, 2D.2). The assignment of 3D surface locations to parts does not require any annotation, since for synthetically rendered images, the actual 2D bounding boxes are known; their automatic subdivision into the same regular grid that was used for the

real training images directly establishes the link between appearance parts and 3D geometry.

5.2. Mixture Models

After perspective projection of a synthetic model, for each image pixel its 3D position on the original 3D model surface is known; as a consequence, the 3D points belonging to each part under each of the specified viewpoints can be determined and projected into a common object coordinate system as shown in figure 1 (3D.3). Training objects can have any surface structure, since pixels not belonging to rendered surface patches are discarded. Figure 4, left and center, displays the 3D point clouds for one car base viewpoint and four parts; different colors indicate different parts. This representation now allows to associate regions of the 3D object surfaces under perspective projection to the corresponding appearance parts, since for the rendered as well as the real images the same regular grid of $V \times W$ parts was used. This link holds true independently of the real image training data on which the appearance has been trained, as long as the same regular grid has been used to determine the part regions in synthetically rendered images and annotated real training images.

Once all available models of one object class have been processed under all discretely sampled viewpoints, Gaussian mixtures are fitted to the point clouds of each part per base viewpoint, using the standard Expectation-Maximization procedure. The choice of Gaussian mixtures reflects a trade-off between a faithful representation of the 3D geometry and a conveniently parameterized formulation which later allows to efficiently evaluate the probability of co-occurrence of parts, given the geometry model. This trade-off is reflected in the number of mixtures to represent each parts' geometry: more mixtures will allow to better represent the geometry, while at the same time increasing computational cost during pose estimation. In this approach, the number of mixtures per part is iteratively chosen according to the MDL criterion. Assuming that each part obeys a multivariate multimodal Gaussian distribution with parameterset $\theta_{k\in\{1\ldots K\}} = (\mu_k, R_k, w_k)$ in 3D (where μ_k is the centroid of each mixture component, R_k its covariance and w_k the weight of the mixture component), for each part distribution X the likelihood

$$p(X|K,\theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} p(x_{(n,3D)}|\theta_k) \qquad (1)$$

of each of the N 3D points' $x_{(n,3D)} \in X$ belonging to mixture k is maximized, while accounting for the MDL penalty term. Figure 4, right, shows the fitted mixture models for the 3D point distribution of the parts of a car from base viewpoint "rear". For each part under each base viewpoint, such a representation of its originating 3D surface positions is built.

6. Detection

This section outlines the detection steps, starting with the initial 2D predetection of the entire object, the detection of pose-specific 2D parts for the most probable base viewpoint, and the maximum-likelihood optimization process to estimate the remaining pose parameters.

6.1. 2D Detection

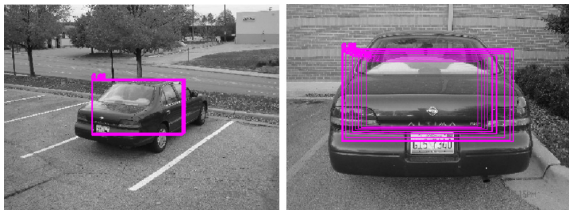

Figure 5. Initial detections with a full object spatial pyramid classifer; note the frequent underestimation of object scale due to the lack of object pose information.

6.1.1 Pre-Detection

The 2D detection process starts with an initial pre-detection to identify regions of interest potentially containing fully or partially visible objects. This method follows the work of [4] in using a sliding-window detection and a subsequent mean-shift mode estimation to merge and localize these regions of interest in image and scale space. This detection step alone is usually unable to generate a reliable localization, since it does not deal with occlusion and is sensitive to the detector window dimensions chosen. In addition, no sliding-window approach can sample all possible window layouts on all possible scales; consequently, additional verification steps are necessary. Figure 5 shows some example detections for the full object pre-detector; note that the pre-detector frequently underestimates the actual scale of the object which is due to its lacking information on the full object pose. In the following steps, knowledge on the full 3D geometry of the object class allows to accurately choose the entire image region containing the object, thereby significantly improving this initial detection and providing an evaluation score which measures the consistency of the detected parts with the learnt geometry model.

6.1.2 Pose-Specific Parts Detection

The part detection forms the fundament which the 3D pose estimation will rely on. Section 4 described how classifiers for different regions of an object under each base viewpoint are computed. Typically trained on much smaller image parts, the discriminativity of these part detectors is reduced; however, by computing them only on the previously identified regions of interest in the test images, much of the background variability is removed which allows to focus

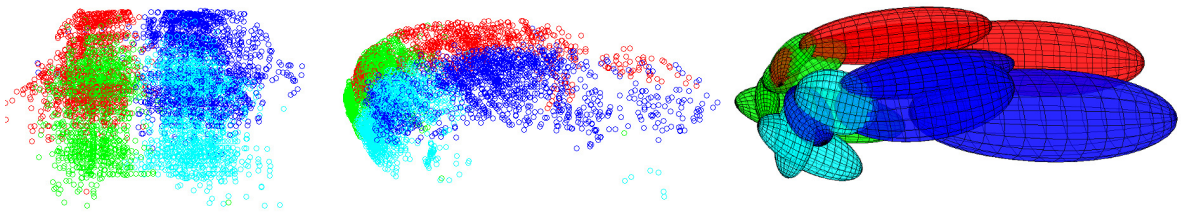

Figure 4. 3D point distributions and fitted mixtures for four parts of the car class from base viewpoint "rear" (left: projection from actual viewpoint, center: rotated, right: estimated mixtures).

the training process on differentiating between base viewpoints and parts on the objects. In addition, these parts can be densely computed on every pixel within the region of interest. The large number of resulting detections increases robustness of the following pose estimation step. A simple voting procedure is used to determine the most likely azimuth-only base viewpoint $\upsilon_i = (1, a_i, 0)$, given all the N detected parts $x_{(n,2D)}$ with their detection probability $p(x_{(n,2D)})$ in the region of interest:

$$p(\upsilon_i) = \sum_{n=1}^{N} p(\upsilon_i | x_{(n,2D)}) p(x_{(n,2D)}). \tag{2}$$

Note that this voting does not yet take the distribution of parts into consideration; it only selects the most promising base viewpoints to evaluate in the subsequent pose estimation. Some part detection results are visualized in figure 6, along with the most likely base viewpoint votes illustrated as histograms.

6.2. 3D Pose Estimation

For all detected parts $x_{(n,2D)}$ of a base viewpoint hypothesis υ_i, an iterative pose estimation now provides an evaluation of the probability of occurrence of the refined viewpoint in simplified spherical camera parameters $\upsilon = (r, a, b)$ as illustrated in figure 2. The required camera parameters are those which maximize the likelihood of the detected 2D parts after perspective projection Φ_υ of the K 3D Gaussian mixtures of this base viewpoint into the image space:

$$\begin{aligned} &\underset{\upsilon \in \Upsilon_i}{\operatorname{argmax}} \prod_{n=1}^{N} p(x_{(n,2D)} | \upsilon) = \\ &\underset{\upsilon \in \Upsilon_i}{\operatorname{argmax}} \prod_{n=1}^{N} \sum_{k=1}^{K} w_k \, \mathcal{N}\left(x_{(n,2D)} | \Phi_\upsilon(\mu_{(k,3D)}), \Phi_\upsilon(R_{(k,3D)})\right) \end{aligned} \tag{3}$$

To simplify the computation of the likelihood under the perspective projection Φ_υ of the per-part covariances $R_{(k,3D)}$ into image space, Φ_υ is approximated by the Taylor expansion localized at the mixture centroids $\mu_{(k,3D)}$, assuming the projection to be locally affine:

$$\Phi_\upsilon(x_{(n,3D)}) \approx \Phi_\upsilon(\mu_{(k,3D)}) + J_{\Phi_\upsilon}(x_{(n,3D)} - \mu_{(k,3D)}) \tag{4}$$

which allows to compute the approximate covariance of the projected 3D mixtures $\Phi_\upsilon(R_{(k,3D)})$ from the original covariances $R_{(k,3D)}$ using the Jacobian J_{Φ_υ} of the projection Φ_υ evaluated at the 3D centroids $\mu_{(k,3D)}$:

$$\Phi_\upsilon(R_{(k,3D)}) \approx J_{\Phi_\upsilon}(\mu_{(k,3D)}) \cdot R_{(k,3D)} \cdot J^t_{\Phi_\upsilon}(\mu_{(k,3D)}). \tag{5}$$

The optimization problem is again solved iteratively in an EM-like fashion under the constraints Υ_i given in section 7 which reflect the discretization used during training; the EM update step is done using a genetic algorithm [10], since it performed best in the experiments due to its robustness towards local optima.

The resulting detection now allows to evaluate the probability of occurence of an object of the searched-for class under a consistent 3D pose. Moreover, the 3D bounding-box backprojected into the image can be used to determine the smallest circumscribed 2D rectangle which significantly improves the 2D scale estimates of the initial detection step. Note, however, that the 3D pose estimation obtained is relative to the virtual camera parameters used to generate the geometry training data from the synthetic model database. Without information on the real camera used to take the specific test image, the computed virtual 3D pose does not relate to the actual metric 3D pose of the object, but provides only orientations and relative distances. Still, if metric calibration data of the camera used to take each test image was available, the virtual camera pose could be promoted to an actual 3D measurement.

7. Experimental Results

On the publicly available 3D Object Category datasets CAR and BICYCLE [15], our approach was evaluated on two tasks. Object detection in 2D was used to assess the contribution of the geometric model with respect to object localization in image and scale space. The accuracy of our approximate pose estimation in addition to the 2D detection was evaluated with respect to groundtruth orientation annotations.

7.1. Dataset

The approach relies on training data from two separate sources. The 3D geometric representation is built from

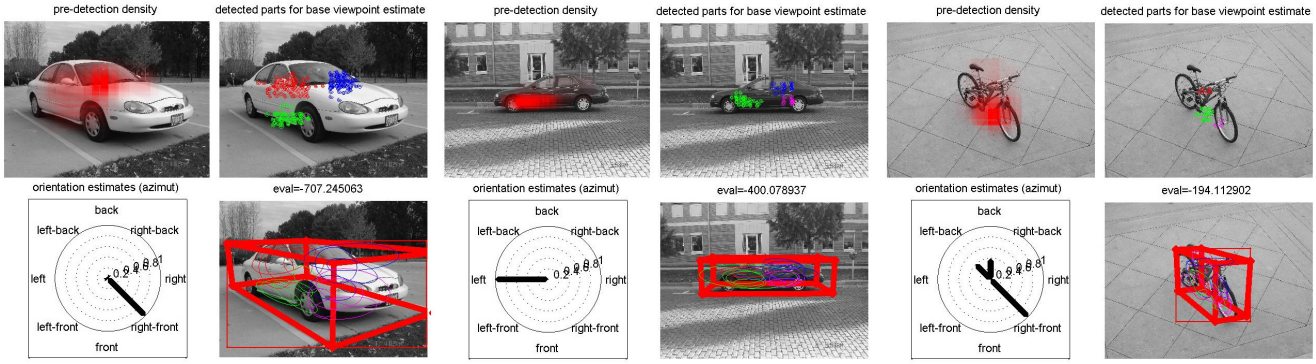

Figure 6. Some results illustrating the complete detection process on the 3D Object Category datasets CAR and BICYCLE [15]. For each result, the predetection density, the detected parts, the base viewpoint votes and the final pose estimation are visualized.

3D models available from the commercial distributors turbosquid.com and doschdesign.com. We used two car and two bicycle models shown in figure 3 which are representative of the object class geometries contained in the test database. For training, the 3D models are normalized to unit scale in the virtual camera coordinate system and rendered from distances $r \in \{1 \cdots 5\}$ (in multiples of 3D model radius), six elevation angles in steps of 10° and all azimuth angles in 22.5° steps. Consequently, given an initial hypothesis υ_i, the pose estimation step is constraint to $\Upsilon_i = \{r \in \{1 \cdots 5\}, a \in \{a_i \pm 22.5^\circ\}, b \in \{0 \cdots 60^\circ\}\}$ where a_i is the azimuth angle associated with the base viewpoint hypothesis.

Appearance training relies on the annotated images available in the 3D Object Category datasets CAR and BICYCLE [15]. For each object class, the dataset contains images of 10 different object instances from 42 different viewpoints. We follow the evaluation protocol described in [15], using the images of 7 randomly selected object instances per class for training and those of 3 unseen instances for testing. Unlike [15] who chose to omit the farthest distance, this approach is trained and evaluated on all viewpoints in the database. 2D training bounding boxes are computed from the provided groundtruth segmentation masks and the base viewpoint classifiers are trained on the available approximate viewpoint annotations, consisting of three distance units (near, medium, far), three elevation units (low, medium, high) and 8 azimuth steps of approximatively 45°. Codebook sizes of 2048 codewords for the full detector on 3 pyramid levels and 1024 codewords on two pyramid levels for the part detector performed best in our experiments.

7.2. Experiments

The 2D localization task is evaluated with the standard 50% VOC Challenge overlap criterion on the axis-aligned rectangular 2D bounding boxes obtained from backprojecting the 3D bounding boxes generated by the pose estimation. To handle rare cases where the pose estimation does not converge, the 2D predetection result serves as a fallback if the overlap between predetection and backprojected pose estimation is below 30%. The precision/recall curve obtained with our 2D detection approach on the CAR dataset is given in figure 7 (AP 76.7%). We compare to the best currently reported pure 2D approach [7] (AP 72.6%) and the most recent 3D approach of [16] (AP 55.3%). As can be seen, our detection approach outperforms the state of the art on the CAR dataset. On the BICYCLE dataset, our method achieves an AP of 69.8% which is slightly below the 2D results reported by [7], probably because on the narrow image regions of bicycle frontal and rear views, the 3D backprojections into 2D image space used by the present approach tend to overestimate relative to the provided groundtruth annotations.

To demonstrate the contribution of our pose estimation component to the 2D detection, we again evaluated the detection task on the same datasets, this time omitting the pose estimation. Instead, the detected 2D parts cast votes for their potential parent objects, similar to an implicit shape model [17]. The scores for both classes, bicycles (AP 63.7%) and cars (AP 59.9%), are significantly below those obtained with our combined detection and pose estimation approach (bicycles (AP 69.8%) and cars (AP 76.7%), see above). By including a pose estimation into the detection process, the detection precision can thus be substantially increased.

In order to benchmark the 3D pose estimation, only the orientation estimations can be compared against the annotated orientations, since the provided elevation and distance groundtruth of the test datasets is too approximate. We bin the continuous orientation estimates in 45° steps to be comparable to the groundtruth annotations. The confusion matrices obtained on the CAR and BICYCLE datasets are shown in figure 8; for cars, the diagonal views suffer from multiple symmetries; for bicycles, front and rear views are more difficult to estimate correctly. On the car dataset, the achieved AP of 70% compares favourably to [16] (approx. 67%); no published pose estimation results on the BICYCLE dataset are currently available for comparison.

Figure 6 shows some examples of the full detection process. In each result window, the predetection density is vi-

sualized in the top left area, the detected parts which contributed to the best base viewpoint are plotted in the top right area. The votes cast for each base viewpoint bin are visualized as histograms in the lower left area. In the bottom right, the pose estimation along with the backprojected covariance ellipses of the parts is given; note that no additional priors on viewpoints or ground planes are used.

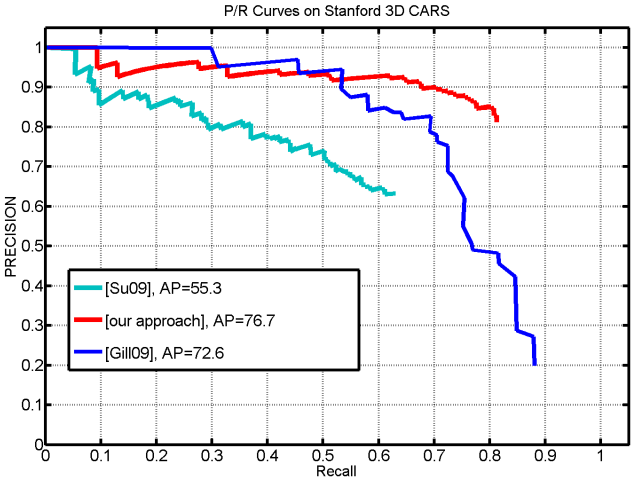

Figure 7. Precision/Recall curves on the 3D Object Category dataset CAR [15].

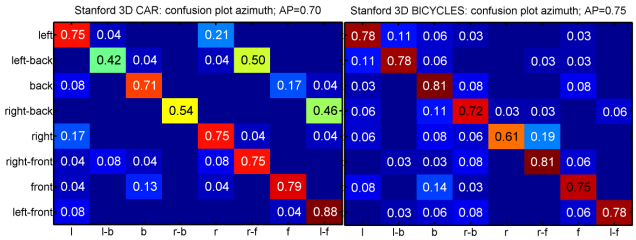

Figure 8. Confusion matrices (rows: groundtruth, columns: estimates) for orientation estimates on the 3D Object Category datasets CAR and BICYCLE [15].

8. Conclusion

This paper has introduced a method for including external 3D geometry from synthetic CAD models into a 2D part-based appearance detection method, yielding an approximate 3D pose estimation and an evaluation score for 3D geometric consistency of 2D part detections. Future work will focus on evaluating the contribution of the pose estimation for large-scale object detection tasks and extending the method to more object classes.

Acknowledgments

The first author acknowledges support by BMBF grant SiVe FKZ 13N10027.

References

[1] M. Arie-Nachmison and R. Basri. Constructing implicit 3D shape models for pose estimation. In *International Conference on Computer Vision*, 2009.

[2] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[3] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Object recognition by combining appearance and geometry. In *Conference on Computer Vision and Pattern Recognition*, 2005.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision and Pattern Recognition*, 2005.

[5] P. Felzenszwalb, D. McAllester, and D. Ramana. A discriminatively trained, multiscale, deformable part model. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[6] P. Flynn and A. Jain. CAD-based computer vision: from CAD models to relational graphs. *Transactions on Pattern Analysis and Machine Intelligence*, 13:114 – 132, 1991.

[7] G. Gill and M. Levine. Multi-view object detection based on spatial consistency in a low dimensional space. In *Symposium of the German Association for Pattern Recognition (DAGM)*, 2009.

[8] B. Heisele, G. Kim, and A. J. Meyer. Object recognition with 3D models. In *British Machine Vision Conference*, 2009.

[9] D. Hoiem, C. Rother, and J. Winn. 3D LayoutCRF for multi-view object class recognition and segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2007.

[10] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. IEEE International Conference on Neural Networks*, volume IV, pages 1942–1948, 1995.

[11] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *Conference on Computer Vision and Pattern Recognition*, 2006.

[12] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D feature maps. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[13] M. Ozuysal, V. Lepetit, and P.Fua. Pose estimation for category specific multiview object localization. In *Conference on Computer Vision and Pattern Recognition*, 2009.

[14] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations. In *IEEE International Conference on Computer Vision*, 2007.

[15] S. Savarese and L. Fei-Fei. View synthesis for recognizing unseen poses of object classes. In *European Conference on Computer Vision*, 2008.

[16] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *IEEE International Conference on Computer Vision*, 2009.

[17] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *Conf. on Computer Vision and Pattern Recognition*, 2006.

[18] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[19] P. Yan, S. M. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *International Conference on Computer Vision*, 2007.